



AMENDMENTS TO THE CLAIMS

Upon entry of this amendment, the following listing of claims will replace all prior versions and listings of claims in the pending application.

IN THE CLAIMS

Please amend claims 1, 2, 3, 12, 13, 14, 23, 25, and 26, and add claims 35-37 as follows:

1. (Currently Amended) A method for invoking a method defined with an object-oriented environment comprising:

retrieving a set of method signatures for a method referenced in a requested method invocation, where each method signature corresponds to a method provided by an object within an object-oriented environment, and further wherein each signature includes a method name and lists any data types of input parameters to be received by the corresponding method;

comparing the data types of input parameters of each method represented by of the signatures to data types of input parameters received from passed by the requested method invocation an array-based computing environment to determine suitability of each method to receive input parameters passed by the requested method invocation;

ranking the method signatures as a function of the comparison;

selecting one of the method signatures according to the ranking; and

invoking, in response to the requested method invocation, the method of the object-oriented computing environment corresponding to the selected method signature.

2. (Currently Amended) The method of claim 1, wherein the requested method invocation is requested by an array-based computing environment is provided by a mathematical tool.

3. (Currently Amended) The method of claim 1, wherein ranking the method signatures comprises calculating a fitness ranking for each signature [[.]], the fitness ranking representative

of a level of suitability of the data types of the input parameters of the method represented by the signature to use the input parameters passed by the requested method invocation.

4. (Original) The method of claim 3, wherein calculating a fitness ranking for each signature includes generating a preference value for each data type of the signature and adjusting the fitness ranking of the corresponding signature as a function of the comparison.

5. (Original) The method of claim 1, wherein, for the signature data types that are superclasses of the data types of the input parameters received from the computing environment, calculating the fitness ranking for each signature includes calculating a difference in level within a class hierarchy for the signature data type and the data type of the corresponding input parameter received from the computing environment.

6. (Original) The method of claim 3, wherein calculating a fitness ranking for each signature includes calculating a difference in a number of dimensions between the signature data type and the input parameter received from the computing environment.

7. (Original) The method of claim 1, wherein comparing each data type of the signature to the data type of the corresponding input parameter includes accessing a data structure storing data types of the object-oriented environment ordered by preference.

8. (Original) The method of claim 6, wherein the data structure is a two-dimensional array storing, along a first dimension, data types supported by the object-oriented environment ranked according to preference, and further wherein a second dimension of the array corresponds to data types supported by the array-based computing environment.

9. (Original) The method of claim 1, wherein invoking the method includes:
converting the input parameters to data types supported by the object-oriented environment; and

converting return values from the method to data types supported by the computing environment.

10. (Original) The method of claim 1, wherein the object-oriented environment includes a virtual machine, and further wherein invoking the method includes interpreting the method via the virtual machine.

11. (Original) The method of claim 8, wherein the virtual machine is a Java virtual machine.

12. (Currently Amended) A computer program, tangibly stored on a computer-readable medium, for invoking a method defined within an object-oriented environment, the computer program comprising instructions operable to cause a programmable processor to:

retrieve a set of method signatures for a method referenced in a requested method invocation, where each method signature corresponds to a method provided by an object within an object-oriented environment, and further wherein each signature includes a method name and a data type for each input parameter received by the corresponding method;

compare the data types of each input parameter of each method represented by the signatures to data types of input parameters ~~received from~~ passed by the requested method invocation ~~an array-based computing environment~~ to determine suitability of each method to receive the input parameters passed by the requested method invocation;

rank the method signatures as a function of the comparison;

select one of the method signatures according to the ranking; and

invoke, in response to the requested method invocation, the method of the object-oriented computing environment corresponding to the selected method signature.

13. (Currently Amended) The computer program of claim 12, wherein the requested method invocation is requested by an array-based computing environment is provided by a mathematical tool.

14. (Currently Amended) The computer program of claim 12, wherein the computer program ranks the method signatures by calculating a fitness ranking for each signature[[.]],the fitness ranking representative of a level of suitability of the data types of the input parameters of the method represented by the signature to use the input parameters passed by the requested method invocation.

15. (Original) The computer program of claim 14, wherein the computer program calculates a fitness ranking for each signature by generating a preference value for each data type of the signature and adjusting the fitness ranking of the corresponding signature as a function of the comparison.

16. (Original) The computer program of claim 12, for the signature data types that are superclasses of the data types of the input parameters received from the computing environment, the computer program calculates the fitness ranking for each signature by calculating a difference in level within a class hierarchy for the signature data type and the data type of the corresponding input parameter received from the computing environment.

17. (Original) The computer program of claim 12, wherein the computer program calculates a fitness-ranking for each signature by calculating a difference in a number of dimensions between the signature data type and the input parameter received from the computing environment.

18. (Original) The computer program of claim 12, wherein the computer program compares each data type of the signature to the data type of the corresponding input parameter includes by

accessing a data structure storing data types of the object-oriented environment ordered by preference.

19. (Original) The computer program of claim 18, wherein the data structure is a two-dimensional array storing, along a first dimension, data types supported by the object-oriented environment ranked according to preference, and further wherein a second dimension corresponds to data types supported by the array-based computing environment.

20. (Original) The computer program of claim 11, wherein the computer program invokes the target method by converting the input parameters to data types supported by the object-oriented environment and converting return values from the method to data types supported by the computing environment.

21. (Original) The computer program of claim 11, wherein the computer program invokes the method by interpreting the target method with a virtual machine.

22. (Original) The computer program of claim 21, wherein the virtual machine is a Java virtual machine.

23. (Currently Amended) A system comprising:

an object-oriented environment operating within a computer, wherein the object-oriented environment includes an interface for identifying methods provided by objects within the object-oriented environment; and

a technical computing environment comprising: a calculation workspace; a command interpreter; and

a signature selector, wherein when the command interpreter encounters within the calculation workspace a requested method invocation comprising a reference to a method implemented by an object defined within the object-oriented environment, the command

interpreter instructs the signature selector to access the interface of the object-oriented environment to retrieve and rank a list of signatures corresponding to the method referenced in the requested method invocation ~~methods defined within the object oriented environment~~, wherein the command interpreter invokes in the object-oriented environment one of the methods represented by one of the signatures as a function of the ranking [[.]], the ranking determining suitability for data types of input parameters of each method represented by the signatures to receive data types of input parameters passed by the requested method invocation.

24. (Original) The system of claim 23, wherein the technical computing environment is provided by a mathematical tool executing on the computer.

25. (Currently Amended) The system of claim 23, wherein the signature selector ranks the method signatures by calculating a fitness ranking for each signature, the fitness ranking representative of a level of suitability of the data types of the input parameters of the method represented by the signature to use the input parameters passed by the requested method invocation.

26. (Currently Amended) The system of claim 25, wherein the signature selector calculates a fitness ranking for each signature by:

comparing the each data type of an input parameter listed by the signature to a data type of a corresponding input parameter received from the requested method invocation ~~an array based computing environment~~; and

adjusting the fitness ranking of the corresponding signature as a function of the comparison.

27. (Original) The system of claim 23, wherein for at least one method signature, the signature selector ranks the method signature by calculating a difference in level within a class

hierarchy for the signature data type and the data type of corresponding input parameter received from the computing environment.

28. (Original) The system of claim 23, wherein the signature selector determines a preference value for each data type included in the method signatures; and further wherein the computer program calculates the preference value of each signature according to the preference values for the data types included in the signature.

29. (Original) The system of claim 23, wherein the signature selector includes a two-dimensional array, wherein along a first dimension the array stores data types supported by the first operating environment ranked according to preference, and further wherein a second dimension corresponds to data types supported by the computing environment.

30. (Original) The system of claim 23, wherein the signature selector includes conversion tables to convert the input parameters to data types supported by the object-oriented environment and to convert return values from the method to data types supported by the computing environment.

31. (Original) The system of claim 23, wherein the object-oriented environment includes a virtual machine for interpreting the invoked method.

32. (Original) The system of claim 31, wherein the virtual machine is a Java virtual machine.

33. (Original) The system of claim 23, wherein the interface is a Java Native Interface (JNI).

34. (Original) The system of claim 23, wherein for at least one method signature, the signature selector ranks the method signature by calculating a difference in a number of dimensions between the signature data type and the input parameter received from the computing environment.

35. (New) The method of claim 1, wherein each signature includes a method name comprising the name of the method in the requested method invocation, and wherein each method represented by the signature corresponds to a method provided by the same object.

36. (New) The computer program of claim 12, wherein each signature includes a method name comprising the name of the method in the requested method invocation, and wherein each method represented by the signature corresponds to a method provided by the same object.

37. (New) The system of claim 23, wherein each signature includes a method name comprising the name of the method in the requested method invocation, and wherein each method represented by the signature corresponds to a method provided by the same object.